IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/705,525

Filing Date: November 10, 2003

Applicant(s):   Attila Barta

Entitled: PRE-DEPLOYMENT COMPONENT HOSTING

                        ENVIRONMENT ANALYZER

Examiner: Qing Chen

Group Art Unit: 2191

Attorney Docket No.:   RSW920030176US1 (7161-517U)

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sirs:

    Submitted herewith is Appellants' Appeal Brief in support of the Notice of

Appeal filed February 10, 2010.

Date: April 12, 2010   Respectfully submitted,

                                         /Steven M. Greenberg/
                                         Steven M. Greenberg
                                         Registration No. 44,725
                                         Customer Number 46320

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/705,525

Filing Date: November 10, 2003

Applicant(s):   Attila Barta

Entitled: PRE-DEPLOYMENT COMPONENT HOSTING

                      ENVIRONMENT ANALYZER

Examiner: Qing Chen

Group Art Unit: 2191

Attorney Docket No.: RSW920030176US1 (7161-517U)

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sirs:

This Appeal Brief is submitted in support of the Notice of Appeal filed
February 10, 2010, wherein Appellants appeal from the Examiner's rejection of
claims 1, 3 through 13 and 41.

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation by assignment recorded on November 10, 2003, at Reel 014700, Frame 0687.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1, 3 through 13 and 41 are pending in this Application. Claims 1, 3 through 13 have been rejected seven times and claim 41 has been rejected twice. Claim 2 was canceled in the amendment dated September 26, 2006 (the "First Amendment"). Claims 14 through 40 were canceled and claim 41 was added in an amendment dated June 19, 2008 (the "Fourth Amendment"). It is from the multiple rejections of claims 1, 3 through 13 and 41 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

Claims 1 and 3 through 13 were amended initially in the First Amendment. Claims 1, 4 and 13 were amended again in an amendment dated January 22, 2007 (the "Second Amendment"). Claims 1, and 3 through 13 were amended yet again in an amendment dated January 15, 2008 (the "Third Amendment"). Claims 1, 9

and 12 were amended even yet again in the Fourth Amendment. Claims 1 and 41 were amended in the amendment dated January 15, 2009 (the "Fifth Amendment"). Finally, claim 41 was amended in the amendment dated July 6, 2009 (the "Sixth Amendment").

V. SUMMARY OF CLAIMED SUBJECT MATTER

With specific respect to claim 1, a computer implemented method for a pre-deployment analysis of software components of an application prior to deployment of the application is provided. The method includes including, in an installation package for the application, a data structure (Par. [0026]) that provides, for each of the plurality of software components from the application, software component deployment dependency data (Par. [0027]), an indication of necessary software components for an operation of each of the plurality of software components being installed (Par. [0028]), and an indication of incompatibility with a previously installed software component. (Par. [0029]) The method also includes loading the installation package into a memory connected to a computer. (Par. [0038]) Finally, the method includes using the computer connected to the storage and a program installed in a memory of the computer so configured by the installation package, to perform the steps of:

(1) determining a first plurality of software components previously installed on a system; (Par. [0038])

(2) determining a second plurality of software components to be installed on the system; (Par. [0038])

(3) accessing a third plurality of software component deployment dependency data;

(4) determining a fourth plurality of software components suitable for parallel installation; (Par. [0038])

(5) determining an order in which the fourth plurality of software components can be grouped for a fifth plurality of parallel installations; (Par. [0039])

(6) accessing a sixth plurality of metadata from the data structure regarding the second plurality of software components to be installed and accessing a seventh plurality of metadata regarding the first plurality of software components previously installed; and (Par. [0041])

(7) analyzing the sixth plurality of metadata to determine an eighth plurality of potential conflicts between the second plurality of software components to be installed and the first plurality of software components previously installed on the system. (Par. [0041])

The pre-deployment analysis allows the second plurality of software components to be installed in parallel and in a sequence of groups. (Par. [0044]) Additionally, an installation time for the application is reduced.

With respect to claim 41, computer implemented method of using a semantic model to increase the efficiency of deployment of an application to a target is provided by maximizing parallel installation of application software components. The method includes accessing the semantic model to obtain dependency information about software components of an application and including a semantic model in an installation package of the application. (Par. [0038]) The method also includes, in response to loading the installation package into a memory connected to a computer, using the computer so configured by the installation package to perform the following steps:

(1) storing a first record of each of a plurality of the software components that is to be deployed in a read file; (Par. [0037])

(2) storing a second record of each of a plurality of previously installed software components in a registry file; (Par. [0037])

(3) when the read file is available to deploy, examining the registry file and accessing the semantic model to obtain a plurality of dependency information indicating a plurality of relationships among the plurality of the software components to be installed in the target and among a plurality of previously installed software components; (Par. [0038])

(4) using the dependency information to group the plurality of the software components into sets of software components with like dependency levels, wherein

5

a first set of software components from amongst the sets of software components

has no dependencies, a second set of software components from amongst the sets

of software components has dependencies only on the first set of software

components, and a third set of software components from amongst the sets of

software components has dependencies only on the first and second sets of

software components; (Par. [0039])

(5) installing the first set of software components in parallel; (Par. [0040])

(6) responsive to completing installation of the first set of software components,

installing the second set of software components in parallel; (Par. [0040])

(7) responsive to completing installation of the second set of software components,

installing the third set of software components in parallel; (Par. [0040])

(8) when a component is installed, updating the registry file; (Par. [0041])

(9) when a conflict is identified, taking an appropriate action; and (Par. [0041])

(10) displaying a progress report by labeling the plurality of the software

components in the semantic model in a selected level of granularity. (Par. [0041])

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 3, 4 and 6 through 13 have been rejected under 35 U.S.C. § 103(a)

as being unpatentable over U.S. Patent No. 6,442,754 to Curtis et al. (Curtis) in

view of U.S. Patent No. 6,725,452 to Te'eni and further in view of U.S. Patent No. 6,675,382 to Foster et al. (Foster).

Claim 5 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Curtis, Te'eni and Foster, and further in view of U.S. Patent No. 6,918,112 to Bourke-Dunphy et al. ("Bourke-Dunphy").

Claim 41 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Curtis in view of Foster.

VII. THE ARGUMENT

THE REJECTION OF CLAIMS 1, 3, 4 AND 6 THROUGH 13 UNDER 35 U.S.C. § 103(A)

On pages 3 through 8 of the Final Office Action dated November 9, 2009 (the "Final Office Action"), Examiner rejects claim 1 for obviousness. As already explained on pages 9 through 11 of the Seventh Amendment, the Manual of Patent Examination Procedure (M.P.E.P.) 2141 sets forth guidelines intended to assist personnel of the United States Patent and Trademark Office in making a proper determination of obviousness under 35 U.S.C. 103, and to provide an appropriate supporting rationale in view recent judicial developments in regard to 35 U.S.C. § 103. Included as part of M.P.E.P. 2141 are the "Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in KSR International Co. v. Teleflex Inc.," 73 Fed. Reg. 57,526 (2007)

(hereinafter the Examination Guidelines). Section III of M.P.E.P. 2141 is entitled

"Rationales To Support Rejections Under 35 U.S.C. 103." Within this section is

the following quote from the Supreme Court: "rejections on obviousness grounds

cannot be sustained by merely conclusory statements; instead there must be some

articulated reasoning with some rational underpinning to support the legal

conclusion of obviousness." KSR Int'l Co.,127 S. Ct. 1727, 1741 (2007) (quoting

In re Kahn, 441 F.3d 977, 988 (Fed. Cir. 2006)).

Referring to Section III of the Examination Guidelines, the following is a list

of rationales that may be used to support a finding of obviousness under 35 U.S.C.

§ 103:

> (A) Combining prior art elements according to known methods to yield predictable results;
> (B) Simple substitution of one known element for another to obtain predictable results;
> (C) Use of known technique to improve similar devices (methods, or products) in the same way;
> (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
> (E) "Obvious to try" - choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
> (F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art;
> (G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

On page 18 of the Final Office Action, Examiner establishes that Examiner has employed rationale (A). As noted in the Seventh Amendment, with respect to rationale (A), the Examination Guidelines set forth a precise process for which the Examiner must follow in order to establish a prima facie case of obviousness under 35 U.S.C. § 103(a). Specifically, to reject a claim based on this rationale, Office personnel must resolve the Graham factual inquiries. Thereafter, Office personnel must then articulate the following:

> (1) **a finding that the prior art included each element claimed**, although not necessarily in a single prior art reference, with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference;
> (2) a finding that one of ordinary skill in the art could have combined the elements as claimed by known methods, and that in combination, each element merely would have performed the same function as it did separately;
> (3) a finding that one of ordinary skill in the art would have recognized that the results of the combination were predictable; and
> (4) whatever additional findings based on the Graham factual inquiries may be necessary, in view of the facts of the case under consideration, to explain a conclusion of obviousness.

It is the position of Appellants that under M.P.E.P. 2141 and the Examination Guidelines set forth therein, Examiner has not adequately articulated a finding that the prior art included each element claimed with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference.

9

In this regard, claims 1 as amended recites a composite service enabling data

processing system. For the convenience of the Honorable Board, claim 5 as

amended is reproduced herein as follows:

1. A computer implemented method for a pre-deployment analysis of a plurality
of software components of an application prior to deployment of the application,
comprising:

including, in an installation package for the application, **a data structure
that provides, <u>for each</u> of the plurality of software components from the
application**, a software component deployment dependency data, an indication of
necessary software components for an operation of each of the plurality of
software components being installed, **and an indication of incompatibility with
a previously installed software component**; and

loading the installation package into a memory connected to a computer;
and,

using the computer connected to the storage and a program installed in a
memory of the computer so configured by the installation package, performing
the steps of:

determining a first plurality of software components previously
installed on a system;

determining a second plurality of software components to be
installed on the system;
accessing a third plurality of software component deployment
dependency data;

determining a fourth plurality of software components suitable for
parallel installation;

determining an order in which the fourth plurality of software
components can be grouped for a fifth plurality of parallel installations;

accessing a sixth plurality of metadata from the data structure
regarding the second plurality of software components to be installed and
accessing a seventh plurality of metadata regarding the first plurality of software
components previously installed; and

analyzing the sixth plurality of metadata to determine an eighth
plurality of potential conflicts between the second plurality of software

10

components to be installed and the first plurality of software components
previously installed on the system;

wherein the pre-deployment analysis allows the second plurality of
software components to be installed in parallel and in a sequence of groups; and

wherein an installation time for the application is reduced.

Integral to claim 1 is the inclusion within the installation package for each software

component of the application, of an indication of incompatibility with a previously

installed software component. This teaching cannot be found in Te'eni.

Appellants maintained as much in the Seventh Amendment. Specifically,

Appellants argued on page 13 of the Seventh Amendment:

> In referring to column 1, lines 61 through 64 of Te'eni, it is clear that only a
> general statement exists that conflicts arise during installation of components.
> Column 5, lines 10 through 17 only add that an upgrade process is created for a
> sequence of tasks that includes collecting information for a "dependency conflicts
> resolving process".

In response, Examiner stated on page 19 of the Final Office Action:

> Note that it is well-known in the art that when performing an upgrade of a
> software program, dependency conflicts may arise among the software
> components present (previously installed software components) and the software
> components to be installed. Te'eni's invention attempts to resolve such
> dependency conflicts by utilizing a virtual upgrade module to collect all the
> information necessary for the dependency analysis and the dependency conflicts
> resolving process (an indication of incompatibility with a previously installed
> software component).

Of import, while at present it is known that when performing an upgrade of a

software program, dependency conflicts may arise, Examiner has provided no

evidence that such knowledge was present in the art at the time of Appellants'

invention. Further, and more importantly, Appellants' claims require much more

than mere knowledge that a conflict may have arisen. Rather, Appellants' claims

require in accordance with the plain claim language of claim 1 "a data structure that

provides, for each of the plurality of software components from the application, ... an

indication of incompatibility with a previously installed software component."

Examiner has not attempted to map each claim term set forth above to Te'eni as

required by M.P.E.P. 2141.

THE REJECTION OF CLAIM 5 UNDER 35 U.S.C. § 103(A)

For the convenience of the Examiner, claim 5 stands or falls together with

claim 1.

THE REJECTION OF CLAIM 41 UNDER 35 U.S.C. § 103(A)

With respect to claim 41, a computer implemented method is provided for the

use of a semantic model to increase the efficiency of deployment of an application

to a target by maximizing parallel installation of application software components.

Claim 41 as previously amended recites:

> 41.   A computer implemented method of using a semantic model to increase
> the efficiency of deployment of an application to a target by maximizing parallel
> installation of application software components, the computer implemented
> method comprising:
>
>   accessing the semantic model to obtain a dependency information about
> software components of an application;

including a semantic model in an installation package of the application;

responsive to loading the installation package into a memory connected to a computer, using the computer so configured by the installation package to perform steps comprising:

storing a first record of each of a plurality of the software components that is to be deployed in a read file;

storing a second record of each of a plurality of previously installed software components in a registry file;

when the read file is available to deploy, examining the registry file and accessing the semantic model to obtain a plurality of dependency information indicating a plurality of relationships among the plurality of the software components to be installed in the target and among a plurality of previously installed software components;

using the dependency information to *group* the plurality of the software components into sets of software components *with like dependency levels*, wherein a first set of software components from amongst the sets of software components has no dependencies, a second set of software components from amongst the sets of software components has dependencies only on the first set of software components, and a third set of software components from amongst the sets of software components has dependencies only on the first and second sets of software components;

installing the first set of software components in parallel;

responsive to completing installation of the first set of software components, installing the second set of software components in parallel;

responsive to completing installation of the second set of software components, installing the third set of software components in parallel;

when a component is installed, updating the registry file;

when a conflict is identified, taking an appropriate action; and

displaying a progress report by labeling the plurality of the software components in the semantic model in a selected level of granularity.

As noted on page 15 of the Seventh Amendment, column 13, lines 7 through 27

and 33 through 37 of Curtis lacks the important teaching of using dependency

information in order to group software components into sets according to <u>like</u> dependency levels. In response, on pages 20 and 21 of the Final Office Action Examiner addressed much of the rejection of claim 41, however, the Examiner failed to address Appellants' arguments in regard to "grouping" software components into sets.

To the extent the Examiner, having considered the foregoing arguments, persists and prepares an Examiner's Answer, Examiner is reminded of Examiner's responsibility under M.P.E.P. 1207.02(A)(1)(9)(e) to map every claim term in claim 41 to the Curtis and Foster references. In this regard, for the convenience of the Examiner the entirety of is provided herein:

> For each rejection under 35 U.S.C. 102 or 103 where there are questions as to how limitations in the claims correspond to features in the prior art even after the examiner complies with the requirements of paragraphs (c) and (d) of this section, the examiner must compare at least one of the rejected claims **feature by feature** with the prior art relied on in the rejection. **The comparison must align the language of the claim <u>side-by-side</u> with a reference to <u>the specific page, line number, drawing reference number, and quotation</u> from the prior art, as appropriate.**

Specifically, Examiner must point out with particularity the precise teaching in Curtis that maps to the claimed elements **<u>using the dependency information to</u>** **<u>_group_ the plurality of the software components into sets of software</u>** **<u>components _with like dependency._</u>**

14

To the extent that the combination of Curtis, Foster and Te'eni fail to include each element claimed in claims 1, 3 through 13 and 41, Examiner has not properly established a prima facie case of obviousness under M.P.E.P. 2141 and rationale (A) of the Examination Guidelines. Appellants, therefore, respectfully solicits the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. § 103(a).

Date: April 12, 2010   Respectfully submitted,

                                        /Steven M. Greenberg/
                                        Steven M. Greenberg
                                        Registration No. 44,725
                                        Customer Number 46320

VIII. CLAIMS APPENDIX

1. (Previously presented) A computer implemented method for a pre-deployment analysis of a plurality of software components of an application prior to deployment of the application, comprising:

including, in an installation package for the application, a data structure that provides, for each of the plurality of software components from the application, a software component deployment dependency data, an indication of necessary software components for an operation of each of the plurality of software components being installed, and an indication of incompatibility with a previously installed software component; and

loading the installation package into a memory connected to a computer; and,

using the computer connected to the storage and a program installed in a memory of the computer so configured by the installation package, performing the steps of:

determining a first plurality of software components previously installed on a system;

determining a second plurality of software components to be installed on the system;

16

accessing a third plurality of software component deployment

dependency data;

determining a fourth plurality of software components suitable for

parallel installation;

determining an order in which the fourth plurality of software

components can be grouped for a fifth plurality of parallel installations;

accessing a sixth plurality of metadata from the data structure

regarding the second plurality of software components to be installed and

accessing a seventh plurality of metadata regarding the first plurality of software

components previously installed; and

analyzing the sixth plurality of metadata to determine an eighth

plurality of potential conflicts between the second plurality of software

components to be installed and the first plurality of software components

previously installed on the system;

wherein the pre-deployment analysis allows the second plurality of software

components to be installed in parallel and in a sequence of groups; and

wherein an installation time for the application is reduced.

2. (Canceled)

3. (Previously presented) The computer implemented method of claim 1, further comprising updating the data structure with an identity of a ninth plurality of recently installed software components.

4. (Previously presented) The computer implemented method of claim 1, further comprising providing a user with a plurality of options for the eighth plurality of potential conflicts.

5. (Previously presented) The computer implemented method of claim 4, wherein a first option includes aborting an installation.

6. (Previously presented) The computer implemented method of claim 4, wherein a second option includes continuing an installation.

7. (Previously presented) The computer implemented method of claim 6, further including, upon the exercise of the second option, recording an entry in a log indicative of a conflict and of a continuation of installation.

8. (Previously presented) The computer implemented method of claim 1, further comprising:

initiating a removal of a software component from the system; and

identifying a tenth plurality of remaining software components which

depend on the software component to be removed.

9. (Previously presented) The computer implemented method of claim 8, further

comprising providing a user with a plurality of options if the tenth plurality of

dependent remaining software components are identified.

10. (Previously presented) The computer implemented method of claim 9, wherein

a first option includes aborting a removal.

11. (Previously presented) The computer implemented method of claim 9, wherein

a second option includes continuing a removal.

12. (Previously presented) The computer implemented method of claim 8, further

comprising:

identifying a first software component previously installed on the system

which is dependent upon a removed software component; and

determining an identity of a second software component upon which the first

software component depends.

13. (Previously presented) The computer implemented method of claim 12, further

comprising:

installing the second software component upon which the first software

component depends; and

creating a dependency link between the first software component and the

second software component.

14-40. (Canceled)

41. (Previously amended) A computer implemented method of using a semantic

model to increase the efficiency of deployment of an application to a target by

maximizing parallel installation of application software components, the computer

implemented method comprising:

accessing the semantic model to obtain a dependency information about

software components of an application;

including a semantic model in an installation package of the application;

responsive to loading the installation package into a memory connected to a

computer, using the computer so configured by the installation package to perform

steps comprising:

storing a first record of each of a plurality of the software components that is to be deployed in a read file;

storing a second record of each of a plurality of previously installed software components in a registry file;

when the read file is available to deploy, examining the registry file and accessing the semantic model to obtain a plurality of dependency information indicating a plurality of relationships among the plurality of the software components to be installed in the target and among a plurality of previously installed software components;

using the dependency information to group the plurality of the software components into sets of software components with like dependency levels, wherein a first set of software components from amongst the sets of software components has no dependencies, a second set of software components from amongst the sets of software components has dependencies only on the first set of software components, and a third set of software components from amongst the sets of software components has dependencies only on the first and second sets of software components;

installing the first set of software components in parallel;

responsive to completing installation of the first set of software components, installing the second set of software components in parallel;

21

responsive to completing installation of the second set of software

components, installing the third set of software components in parallel;

when a component is installed, updating the registry file;

when a conflict is identified, taking an appropriate action; and

displaying a progress report by labeling the plurality of the software

components in the semantic model in a selected level of granularity.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.